明 細 書

実行装置

技術分野

[0001] 本発明は、プログラムの盗聴および改竄を防止する技術に関し、特に、プログラム実行時に、プログラムが盗聴および改竄されることを防止する技術に関する。

背景技術

[0002] 近年、様々なアプリケーション・プログラム(以下、「アプリケーション」という。)が、パソコンをはじめ、デジタルテレビや携帯電話機などの情報処理機能を有した装置で実行されている。

これらのアプリケーションは、元々その装置に組み込まれていたものや、ユーザが購入し乗せたもの、また、プログラム配信サービスによって提供されているものなど様々である。プログラム配信サービスによるものとは、例えば、インターネットを介してダウンロードしたり、デジタル放送の放送波に多重化されて送信されてくるものである。

[0003] このように、現在、アプリケーションを実行できる機能を有している装置(以下、「実行装置」という。)では、多種多様な用途・目的のプログラムが実行されていることになる。

その一方で、プログラムが改竄されたり、データが盗まれたりという不正行為が起こっている。このような不正行為は、プログラム配信サービス等の円滑な実施の障害となることから、プログラムの改竄等を防止する技術が開発されている。

[0004] 例えば、CPU(中央処理装置:Central Processing Unit)の外部のメモリには、暗号化したプログラムを置くこととし、CPUに読込む際に復号するという技術である(特許文献1、図11参照)。この方法であれば、デバッガ等を用いてメモリを参照したり、メモリの内容を書き換えてプログラムの動作を追ったりという方法によるプログラムの解析が非常に困難となり、結果としてプログラムの改竄やデータの盗聴を防止できる可能性が増大することになる。

特許文献1:特開平2－297626号公報

発明の開示

発明が解決しようとする課題

[0005] しかし、メモリ上に置くプログラムの全てを暗号化し、CPUが必要とする都度に復号化処理を行い、メモリに書き出す都度に暗号化処理を行うため、実行速度が遅くなってしまうという問題がある。

そこで、本発明は、改竄されたり、盗まれたりしては困るようなデータ等の改竄・盗聴を防止しつつ、プログラム実行速度の低下が気にならないような実行装置の提供を目的とする。

課題を解決するための手段

[0006] 上記課題を解決する為に、本発明の実行装置は、オブジェクト指向言語で作成されたアプリケーションプログラムを実行する実行装置であって、前記アプリケーションプログラムは、1以上のメソッドを有する1以上のクラスと秘匿の必要性の有無を表す秘匿性情報とを含んでおり、前記秘匿性情報に基づいて、暗号化が必要であるか否かを判断する暗号化判断手段と、前記メソッドを実行する際に、前記メソッドが操作するデータを含むオブジェクトをメモリに記録するオブジェクト記録手段とを備え、前記暗号化判断手段で暗号化が必要であると判断された場合においては、前記オブジェクト記録手段は、暗号化されたデータを含むオブジェクトを記録することを特徴とする。

発明の効果

[0007] 本発明に係る実行装置は、上述の構成を備えることにより、アプリケーションが実行されている間、メインメモリに記録されるデータのみを暗号化しておくことができるので、暗号化等に必要な処理時間を減らし、かつ、アプリケーションで扱うデータの盗聴・改竄を困難にすることができるようになる。

すなわち、プログラムは通常、命令部とデータ部とで構成されるが、保護したいものがデータである場合は、そのデータのみを暗号化すればよい。その結果として、命令部にアクセスする場合には、暗号化復号化の処理時間が不要になり、全体として保護したいものは保護しつつ、処理時間も最小限度に抑えることが出来るようになる。

[0008] ここで、秘匿性情報が示す秘匿の必要性の有無の対象は、アプリケーション全体という単位、クラス単位、メソッド単位、さらにはフィールド単位などであってもよい。

また、前記秘匿性情報は、更に、秘匿の度合いを示す情報を含み、前記実行装置は、更に、前記秘匿性情報に基づいて暗号化方式を決定する暗号方式決定手段を備え、前記暗号化判断手段で暗号化が必要であると判断された場合においては、前記オブジェクト記録手段は、前記暗号方式決定手段で決定された暗号化方式で暗号化されたデータを含むオブジェクトを記録することとしてもよい。

[0009] これにより実行装置は、アプリケーションの秘匿の度合いによって、暗号化のアルゴリズムを換えることができるので、アプリケーションの重要度とアプリケーションの実行速度とを考慮して、暗号化方式を決めることが出来るようになる。

ここで、秘匿性情報は、暗号化をすべきか否かの情報のみならず、どのようなレベルで保護すべきか、また、どのようなアルゴリズムを使用すべきか、どのくらいの長さの暗号鍵を使用すべきか等の情報を示すものであってもよい。

[0010] また、前記オブジェクト内のデータを書き換えるときは、前記データが暗号化されている場合は、暗号化されたデータを記録することとしてもよい。

また、前記メモリに記録されたオブジェクトは、オブジェクト内のデータが暗号化されているか否かを示す情報を有し、前記情報がオブジェクト内のデータが暗号化されていることを示している場合は、暗号化されたデータを記録することとしてもよい。

[0011] これにより実行装置は、アプリケーションのデータを書き換える際にも、データを暗号化することができるので、アプリケーション実行中は、データが保護されるようになる。

また、前記実行装置は、更に、データが他のデータの所在位置特定に必要なデータであるか否かを判別する判別手段を備え、前記判別手段が、他のデータの所在位置特定に必要なデータであると判別したデータの場合は、暗号化を抑止することとしてもよい。

[0012] これにより実行装置は、暗号化するデータのうち、いわゆる参照型であるデータは暗号化しないことができるので、ガーベージコレクションの際の処理速度の低下を抑えることができるようになる。

すなわち、参照型のデータそのものは暗号化せずとも、その参照先のデータを暗号化しておけば、データを保護する目的は達せられるからである。また、ガーベージコ

レクションの際に参照型のデータを書き換える場合には、暗号化復号化の処理が不要になることから、処理速度の向上につながる。

[0013] また、本発明の実行装置は、プログラムを実行する実行装置であって、前記プログラムは、データ部と秘匿の必要性の有無を表す秘匿性情報とを含んでおり、前記秘匿性情報に基づいて、暗号化が必要であるか否かを判断する暗号化判断手段と、前記実行可能プログラムを実行する際に、前記データ部をメインメモリにロードするロード手段とを備え、前記ロード手段は、前記暗号化判断手段で暗号化が必要であると判断された場合は、暗号化してデータ部をロードすることを特徴とする。

[0014] 本発明に係る実行装置は、上述の構成を備えることにより、プログラムが実行されている間、暗号化が必要なデータ部のみを暗号化してメインメモリにロードすることができるので、暗号化等に必要な処理時間を減らし、かつ、プログラムで扱うデータの盗聴・改竄を困難にすることができるようになる。

図面の簡単な説明

[0015] [図1]本発明にかかる実行装置の構成を表す図である。

[図2]仮想マシン2000の構成を示す機能ブロック図である。

[図3]暗号化度情報2810の構成および内容例を示す図である。

[図4]アプリケーションファイル1000に入っているアプリケーションの構成および内容例を示す図である。

[図5]ローダ2200によってメソッド領域2600に生成されるクラス情報1210の構成および内容例を示す図である。

[図6]暗号アルゴリズム情報2820の構成および内容例を示す図である。

[図7]図7(a)は、オブジェクト2510の構成例を示す図であり、図7(b)は、オブジェクト2510の内容例を示す図である。

[図8]実行装置3000の処理を表すフローチャートである。

[図9]アプリケーション登録処理を表すフローチャートである。

[図10]オブジェクトの生成処理を表すフローチャートである。

[図11]従来技術の構成を表す図である。

符号の説明

[0016]
1000 アプリケーションファイル
1200 アプリケーションクラス
1210 クラス情報
1300 データファイル
1400 メタデータ
1400 メタ情報
1410 起動クラス名
1420 セキュリティ強度情報
1610 対応処理情報
2000 仮想マシン
2100 アプリ登録部
2110 暗号化判定部
2200 ローダ
2210 クラスローダ
2220 ベリファイヤ
2230 JITコンパイラ
2300 インタプリタ
2310 復号化部
2320 暗号化部
2400 ヒープ管理部
2500 ヒープ領域
2510 オブジェクト
2600 メソッド領域
2700 ネイティブクラスライブラリ
2800 暗号化情報記憶部
2810 暗号化度情報
2820 暗号アルゴリズム情報
3000 実行装置

3100 アプリケーション
3200 アプリケーション制御部
3210 アプリ取得プログラム
3300 OS
3400 CPU
3500 ROM
3600 RAM

発明を実施するための最良の形態

[0017] ＜概要＞

本発明に係る実行装置は、実行装置で実行されるアプリケーション毎にその用途・目的が異なることから、アプリケーション毎にセキュリティ要件が異なること、および、アプリケーション自体よりもそのアプリケーションの扱うデータに秘匿性が高い場合が多いことに着目したものである。

[0018] ここで、セキュリティ要件が異なるとは、アプリケーションによって、保護したいデータが異なっており、またさらに、秘匿性の度合いが異なっているということである。

すなわち、データの秘匿性を必要としないアプリケーションを実行する場合には、実行速度の低下を伴う暗号化、復号化の処理は極力減らしたほうが望ましく、また、非常に秘匿性の高いデータを取り扱うアプリケーションを実行する場合には、多少実行速度が遅くなっても、可能な限り強度の高い暗号アルゴリズムを用いてデータを保護したほうがよいこともあるということである。

[0019] 本発明は、扱うデータの重要度に応じて、自由に暗号化の度合いを変えることが可能であることを特徴とするものである。

アプリケーションで扱うデータの秘匿性の度合いは、そのアプリケーションを作成するときに、作成者またはそのユーザのみが知りうるものである。

従って、本発明に係る実行装置は、データを直接アクセス等するプログラムが、そのデータを生成したアプリケーションを特定する機能を有する。

[0020] すなわち、生成したアプリケーションを特定することで、秘匿性の高いデータを扱うプログラムが生成したデータであるか否かを判別することができることになり、秘匿性

の高いデータのみを暗号化することができるようになる。

結果として、本実行装置は、暗号化による実行時間の低下を最小限に抑えつつデータの盗聴を防ぐデータ保護機能を備えているものであるといえる。

[0021] またさらに、アプリケーションに求められる秘匿性の度合いに応じて、暗号アルゴリズムの強度や鍵を選択できるものでもある。

以下、本発明の実施形態に係る実行装置について説明する。

本実施形態では、Javaバーチャルマシン上で動くJavaアプリケーションについて説明する。

[0022] ＜構成＞

図1は、本発明にかかる実行装置の構成を表す図である。

実行装置3000は、アプリケーション3100、仮想マシン2000、アプリケーション制御部3200、OS（Operating System）3300、CPU（Central Processing Unit）3400、ROM（Read Only Memory）3500およびRAM（Random Access Memory）3600で構成される。

[0023] 実行装置3000は、アプリケーションを実行する機能のほか、それぞれの装置特有の機能を有する（図示していない）。本実行装置3000は、具体的には、デジタルテレビ、セットトップボックス、DVDレコーダー、Blu－RayDisc（BD）レコーダー、カーナビ端末、携帯電話、PDAなどの、Java（登録商標、以下同様。）仮想マシンを搭載する電子機器全般が該当する。

[0024] ここで、本実行装置3000のアプリケーションを実行する機能とは、通常のパーソナルコンピュータやデジタル家電機器等に搭載されているソフトウェア実行手段と同様のものである。例えば、実行装置3000がデジタルテレビであれば、受信したデジタルデータを画像に変換し表示するアプリケーションを実行することになる。

まず、アプリケーション3100は、本実行装置3000で実行するアプリケーションであって、本装置外のアプリケーションファイル1000からダウンロードしたものである。このアプリケーションファイル1000には、暗号化されたJavaアプリケーションが入っているものとする。

[0025] 仮想マシン2000は、Java言語で記述されたプログラムを逐次解析し実行するJava

バーチャルマシンである。言い換えれば、ソフトウェアプログラムである仮想マシン2000が、仮想のCPUをシミュレートして、Javaの命令コードを解析実行する。

本実施形態での仮想マシンは、バイトコードをCPU3400が理解可能な実行形式に翻訳するJITコンパイラと呼ばれる機能を持つものとする。

[0026] すなわち、Java言語で記述されたソースプログラムは、バイトコードコンパイラによってバイトコードに変換される。このバイトコードとは、ハードウェアに依存しない中間コードである。このバイトコードが、アプリケーションファイル1000に入っているものとする。

本実施形態の仮想マシンは、このバイトコードを読出し、JITコンパイラで実行形式に翻訳したものを、メモリにローディングする。

[0027] 尚、Javaバーチャルマシンは、一部もしくは全部のバイトコードを直接実行可能なプロセッサと、プロセッサでは直接実行できないバイトコードを実行するインタプリタから構成されるものなど、様々な構成がある(書籍「Java Language Specification(ISBN0－201－63451－1)」等参照)。

また、アプリケーション制御部3200は、アプリケーション3100をダウンロードしたり、仮想マシン2000を起動させるなどの、アプリケーションの実行に必要な処理を実行、制御する機能を有する。

[0028] OS3300は、他のサブプログラムを平行して実行するカーネル及び、ライブラリで構成される技術の総称であり、仮想マシン2000をサブプログラムとして実行する。例えば、Linux等がある。

CPU3400は、仮想マシン2000、OS3300、アプリケーション3100等を実行する機能を有する。

[0029] RAM3600は、具体的にはSRAM(Static Random Access Memory)、DRAM(Dynamic Random Access Memory)等の一次記憶メモリで構成され、CPU3400が処理を行う際、一時的にデータを保存するために使用される。

ROM3500は、具体的にはフラッシュメモリーやハードディスク等の不揮発性メモリで構成され、CPU3400から指示されたデータやプログラムを記憶する。

[0030] 図2は、仮想マシン2000の構成を示す機能ブロック図である。

仮想マシン2000は、アプリ登録部2100、ローダ2200、インタプリタ2300、ヒープ管理部2400、ヒープ領域2500、メソッド領域2600、ネイティブクラスライブラリ2700および暗号化情報記憶部2800で構成される。

また、アプリ取得プログラム3210は、アプリケーション制御部3200のプログラムの1つであり、Java言語で記述されており、アプリケーションをアプリケーションファイル1000からダウンロードする機能を有する。ダウンロードするアプリケーションの中身については、図4を用いて後で説明する。

[0031] まず、仮想マシン2000の、各機能部について説明する。

アプリ登録部2100は、アプリ取得プログラム3210から依頼を受け、登録依頼されたアプリケーションの暗号化の必要性の有無、暗号化する場合は、その方法等を決定し、暗号化情報記憶部2800に記憶させる機能を有する。また、登録依頼されたアプリケーション用のクラスローダオブジェクトを作成する機能を有する。

[0032] アプリ登録部2100は、暗号化判定部2110を備え、暗号化判定部2110が、アプリ取得プログラム3210が取得したアプリケーションの暗号化の必要性の有無の判断と、その暗号化レベルを実行装置3000が実現できるか否かを判定する。

暗号化判定部2110が、アプリケーションの暗号化が可能であると判断したら、アプリ登録部2100は、そのアプリケーションが生成するデータを暗号化する為の暗号アルゴリズムと暗号鍵を決定し、決定した暗号アルゴリズムなどを対応づけて、暗号化情報記憶部2800に記憶させる。

[0033] ローダ2200は、アプリケーションファイル1000やネイティブクラスライブラリ2700などから、クラスファイルをメソッド領域2600にロードする機能を有する。クラスファイルについては、図5を用いて後で説明する。

ローダ2200は、クラスローダ2210、ベリファイヤ2220、JITコンパイラ2230を備えている。

[0034] クラスローダ2210は、クラスファイルをアプリケーションファイル1000から読込みロードする機能を有する。また、クラスローダ2210は、クラスをアンロードする機能も有する。実行が終了し不要になったクラスを仮想マシン2000から取り除く機能である。

次に、ベリファイヤ2220は、クラスのデータ形式の不備や、クラスに含まれるバイト

コードの安全性を判定する機能を有する。ローダ2200は、ベリファイヤ2220において妥当ではないと判定されたクラスはロードをしない。

[0035] JITコンパイラ2230は、バイトコードをCPU3400が理解可能な実行形式に翻訳する機能を有する。

次に、インタプリタ2300は、ローダ2200によりロードされたバイトコードを解釈、実行する機能を有し、Java仮想マシンにおいて中核な処理を行う。

このインタプリタ2300は、復号化部2310を備え、復号化部2310は、ヒープ領域2500のデータを読み出す際に、データが暗号化されている場合に処理前に復号する機能を有する。また、暗号化部2320を備えており、この暗号化部2320は、ヒープ領域2500に記憶するデータを暗号化する場合に、データを書込む前に暗号化する機能を有する。

[0036] ヒープ管理部2400は、インタプリタ2300の制御の下、ヒープ領域にオブジェクトを作成し、また、消去する機能を有する。

さらに、ヒープ管理部2400は、ガベージコレクションを行う機能も有する。ガベージコレクションとは、アプリケーション実行において不要になったワーキングメモリを開放し、他の用途に再利用できるようにする機能である。

[0037] ここで、ヒープ領域2500は、オブジェクトの作成されるメモリをいい、メソッド領域2600は、クラスファイルをロードする先のメモリをいう。これらは、RAM3600に作成される。

また、ネイティブクラスライブラリ2700は、Javaアプリケーションから呼び出されるライブラリであって、OS3300や、実行装置3000が備えているハードウェア、サブプログラム等で提供される機能をJavaアプリケーションへ提供している。

[0038] 暗号化情報記憶部2800は、暗号化に必要な情報を記憶しておく機能を有し、RAM3600に作成される。

また、本実行装置3000は、通常のJavaバーチャルマシンが有する、スレッドを管理する機能部や、スタック領域など(図示していない。)を有するものとする。

尚、実行装置3000の各機能は、実行装置3000のメモリ又はハードディスクに格納されているプログラムをCPUが実行することにより実現される。

[0039] ＜データ＞

以下、本実行装置で用いる主なデータについて、図3から図6を用いて説明する。

図3は、暗号化度情報2810の構成および内容例を示す図である。

この暗号化度情報2810は、暗号化情報記憶部2800に記憶されているものである。

[0040] 暗号化度情報2810は、セキュリティ強度2811およびデータ暗号化強度1812で構成される。

セキュリティ強度2811は、アプリケーションで扱うデータのセキュリティの高さを示すものであり、例では;「0」から「2」の3段階としている。この「0」から「3」までの値を、アプリケーション毎に指定することになる。

[0041] また、データ暗号化強度1812は、データを暗号化する場合の度合いを表す。例えば、セキュリティ強度2811「0」の場合のデータ暗号化強度2812は、暗号化は「不要」である。すなわち、暗号化は行わないことを示す。セキュリティ強度2811「1」以上はデータの暗号化が必要であることを意味している。さらにセキュリティ強度2811「2」以上では、セキュリティ強度2811「1」よりも強力な暗号アルゴリズム、または長い鍵を用いて暗号化を行う必要があることを表している。

[0042] 次に、図4はアプリケーションファイル1000に入っているアプリケーションの構成および内容例を示す図である。

アプリケーション1001は、アプリケーションクラス1200と、データファイル1300と、メタデータ1400を含んでいる。

アプリケーションクラス1200は、アプリケーションを構成する1つ以上のクラスファイルの集合である。

[0043] データファイル1300は、アプリケーションが実行時に使用するデータである。具体的には、画像ファイルや音声ファイルなどが該当する。

また、メタデータ1400には、アプリケーション1001に関する様々な情報が保持されている。例えば、メタデータ1400は、起動クラス名1410とセキュリティ強度情報1420とを含んでいる。

[0044] 起動クラス名1410は、アプリケーションクラス1200のうち、最初に実行されるべきク

ラスの名前である。

セキュリティ強度情報1420は、アプリケーションクラス1200を実行する際に、仮想マシン2000に求められるセキュリティの強さを表す。このセキュリティ強度情報1420で表される情報は、暗号化度情報2810のセキュリティ強度2811と同じものである。

[0045] このセキュリティ強度情報1420の指定の仕方は、例えばバイトコードコンパイラのオプションとして設定することとする。

図5は、ローダ2200によってメソッド領域2600に生成されるクラス情報1210の構成および内容例を示す図である。

クラス情報1210は、クラス名1211、親クラス1212、インタフェーステーブル1213、メソッドテーブル1214、フィールドテーブル1215、クラスローダID1216およびセキュアフラグ1217などで構成される。

[0046] クラス名1211は、当該クラスのクラス名である。

親クラス1212は、当該クラスの親クラスを表す内部形式への参照である。

ここで、参照とは、ポインタやインデックスなど、そのデータの実体を指し示す表現をいう。

インタフェーステーブル1213は、当該クラスが実装するインタフェースへの参照である。

[0047] メソッドテーブル1214は、当該クラスが備えるメソッドの一覧である。

フィールドテーブル1215は、このクラスが備えるフィールドの一覧である。

また、クラスローダID1216は、当該クラスをロードしたクラスローダオブジェクトを表す。具体的には、クラスローダID2821が入っている(図6参照)。

セキュアフラグ1217は、当該クラスがセキュアクラスか否かを表す。

[0048] ここで、セキュアクラスとは、セキュリティ強度情報1420に、セキュリティ強度2811「1」もしくは「2」を指定されたアプリケーション1001のアプリケーションクラス1200に含まれるクラスをいう。

このセキュアフラグ1217が「ON」になっている場合は、このクラスで扱われるデータは、暗号化が必要と判断されることになる。

[0049] また、このセキュアフラグ1217は、クラスがロードされるときに、ローダ2200によって

設定される。本実施形態では、アプリケーション1001のセキュリティ強度情報1420によって、クラスのセキュアフラグ1217が一律に決まることとする。すなわち、暗号化が必要なアプリケーションのすべてのクラスのセキュアフラグ1217は「ON」とする。

[0050] 図6は、暗号アルゴリズム情報2820の構成および内容例を示す図である。

この暗号アルゴリズム情報2820は、暗号化情報記憶部2800に記憶されており、アプリ登録部2100によって、作成される。

暗号アルゴリズム情報2820は、クラスローダID2821、クラスローダアドレス2822、暗号アルゴリズム2823および暗号鍵2824で構成される。

[0051] クラスローダID2821は、アプリ登録部2100がクラスローダに一意に割り当てる識別子である。すなわち、アプリケーション1つに対して、1つのクラスローダ2210が存在することになる。本実施形態では、「0」から昇順につけていくものとする。

クラスローダアドレス2822は、仮想マシン2000から登録を要求されたクラスローダオブジェクトのアドレスである。

[0052] 例では、クラスローダIDが「0」のクラスローダでロードされたアプリケーションが実行時に生成するデータは、暗号化されないことを表し、クラスローダIDが「2」のクラスローダでロードされたアプリケーションが実行時に生成するデータは、「AES(Advanced Encryption Standard)」アルゴリズムで暗号化され、その鍵は「YYYY」であることを表している。

[0053] <動作>

以下、本発明に係る実行装置3000の動作について図7～図10を用いて説明する。

図7(a)は、オブジェクト2510の構成例を示す図であり、動作の説明で、必要に応じて参照する。まず、図7について説明する。

オブジェクト2510は、クラス内のメソッドが実行されるときに、作成されるものであり、ヒープ領域2500に作成される。

[0054] オブジェクト2510は、オブジェクトヘッダ2511とオブジェクトデータ2512とで構成され、オブジェクトヘッダ2511には、当該オブジェクトが属するクラス情報2551、オブジェクトデータ2512のサイズであるデータサイズ2552、オブジェクトデータ2512

が暗号化されているか否かを示す暗号化フラグ2553などが含まれている。この暗号化フラグ2553が「ON」であれば、オブジェクトデータ2512が暗号化されていることとなる。

[0055] この暗号化フラグ2553は、このメソッドが属するクラス、すなわちクラス情報2551で参照されるクラスのセキュアフラグ1217が「ON」の場合、「ON」とする。

クラス情報2551は、クラスローダによって、メソッド領域2600内に生成されたクラスの内部表現のアドレスである。クラス情報1210(図5参照)の先頭アドレスが、クラス情報2551に入っていることになる。

[0056] また、オブジェクトデータ2512は、Javaアプリケーションが動作することによって生成される実行時データであり、0個以上のフィールドを持つ。フィールドの数はオブジェクトが属するクラスにより一意に決定される。

フィールドは、基本型と呼ばれる数値や文字を扱うフィールドと、参照型と呼ばれる他のオブジェクトへの参照を表すフィールドの2種類がある。本実施形態では、オブジェクトデータ2512全体で、暗号化がなされるものとする。

[0057] 図7(b)は、オブジェクト2510の内容例を示す図である。

例えば、クラス情報2551はアドレス「0xdeadbeef」にある内部表現を参照し、オブジェクトデータのデータサイズ2552は「24」である。暗号化フラグ2553の値が「1(ON)」であることで、このオブジェクトが暗号化されていることを表している。

[0058] アドレス「0xdeadbeef」にある内部表現から、クラスローダID1216(図5参照)を参照することで、このクラスをロードしたクラスローダオブジェクトを特定できる。このクラスローダオブジェクトのIDであるクラスローダID1216が「2」であるとすると、そのクラスローダID1216「2」をキーに、暗号アルゴリズム情報2820(図6参照)を検索する。暗号アルゴリズム情報2820のクラスローダID2821が「2」であるので、該当のオブジェクトデータ2512は、「AES」アルゴリズムで、鍵「YYYY」を用いて暗号化されていることが分かる。

[0059] 実際の暗号化は、インタプリタ2300の暗号化部2320が行い、復号化は、インタプリタ2300の復号化部2310が行う。

以下、図8から図10を用いて、実行装置の処理を説明する。

本実施形態では、実行装置の電源を入れた場合、予め定められているアプリケーションが実行されるものとする。

[0060] 図8は、実行装置3000の処理を表すフローチャートである。

まず、ユーザが実行装置3000の電源を投入する(ステップS810)。

通電したCPU3400は、OS3300を起動する(ステップS820)。

起動したOS3300は、仮想マシン2000を起動して(ステップS830)、アプリ取得プログラム3210を起動するよう指示する。

[0061] 指示を受けた仮想マシン2000は、アプリ取得プログラム3210を起動する(ステップS840)。

仮想マシン2000によって起動されたアプリ取得プログラム3210は、アプリケーションファイルからアプリケーション1001を読込み、アプリ登録部2100に、アプリケーションの登録処理を依頼する。ここでは、アプリケーション1001内の、メタデータ1400(図4参照)のみを読込み、アプリ登録部2100に渡す。他のアプリケーションクラス1200とデータファイル1300は、アプリケーションの実行に応じて、適時読込まれることになる。

[0062] このアプリケーション登録処理において、アプリケーション1001の秘匿性の度合いが判定され、暗号化のアルゴリズムなどが決定される。また、このアプリケーション1001用のクラスローダオブジェクトが生成される(ステップS850)。すなわち、このアプリケーション登録処理が終了すると、図6で示す暗号アルゴリズム情報2820に、該当するアプリケーションの暗号化の情報が登録されていることになる。アプリケーション登録処理の詳細については、図9を用いて後で説明する。

[0063] アプリケーションの登録処理を終了したアプリ登録部2100は、その旨をアプリ取得プログラム3210に返す。登録処理が終了した旨を受け取ったアプリ取得プログラム3210は、インタプリタ2300に起動クラス名1410を通知し、ロードを依頼する。

インタプリタ2300は、クラスローダオブジェクトに、起動クラス名1410(図4参照)で指定されたクラス(以下、「起動クラス」という。)をロードする旨指示する。クラスローダオブジェクトは、指定されたクラスをメソッド領域2600にロードする(ステップS860)。この時点で、図5に示すようなクラス情報1210がメソッド領域2600に作成されており

、クラスローダID1216とセキュアフラグ1217が設定されていることになる。

[0064] この際、ベリファイヤ2220により正当性がチェックされ、JITコンパイラ2230により、ネイティブコードに変換されている。

これで、アプリケーションは仮想マシン2000へロードされ、実行可能な状態となる。

クラスローダオブジェクトによる起動クラスのロードが完了すると、インタプリタ2300は、起動クラスのメソッドを実行することでアプリケーションの起動を行なう。

[0065] メソッドを実行するとは、すなわち、ヒープ管理部2400に依頼して、オブジェクトをヒープ領域2500に作成(ステップS870)し、メソッドを実行していくことになる(ステップS880)。

新たなメソッドを実行する場合には、必要に応じて属するクラスをロードし(ステップS860)、オブジェクトを作成し(ステップS870)、メソッドを実行する。

[0066] オブジェクトをヒープ領域2500に作成した時点で、暗号化が必要である場合には、オブジェクトデータ2512が暗号化されており、暗号化フラグが設定されている。

従って、このオブジェクトのフィールドにアクセスする場合には、オブジェクトヘッダ2511の暗号化フラグ2553を参照して、「ON」ならば、読み出すときは、読み出したデータを復号して処理し、また、書込むときは暗号化したデータを書込むことを行う。暗号化フラグ2553が「OFF」ならば、復号化、暗号化は行わずに、フィールドにアクセスする。暗号アルゴリズムなどの取得方法は、図7(b)での説明の通りである。

[0067] 仮想マシン2000を構成する全ての機能部は、オブジェクトに対して読み込みの操作を行う前に、オブジェクトヘッダ2511内の暗号化フラグ2553を検査し、暗号化されている場合には復号化部2310によりそれを復号化し、読み込む。また、オブジェクトに対して書き込みの操作を行う前に、オブジェクトヘッダ2511内の暗号化フラグ2553を検査し、オブジェクトが暗号化されている場合には暗号化部2320により暗号化したデータを書き込む。

[0068] ＜1．アプリケーション登録処理＞

図9を用いて説明する。図9は、アプリケーション登録処理を表すフローチャートである。

仮想マシン2000を起動したアプリ取得プログラム3210は、アプリケーションファイ

ルからアプリケーション1001を読込み、アプリ登録部2100に、アプリケーションの登録処理を依頼する(ステップS910)。

[0069] 登録処理を依頼されたアプリ登録部307は、登録を依頼されたアプリケーションに含まれるセキュリティ強度情報1420(図4参照)を読み出す(ステップS920)。この時、アプリケーションが暗号化されていれば、セキュリティ強度情報1420の読み出しに先立ち復号する。

次に、読み出したセキュリティ強度情報1420に対応する暗号化機能を、実行装置3000が備えているか否かを判定する(ステップS930)。この判定は、暗号化判定部2110に依頼して行う。

[0070] 具体的には、セキュリティ強度情報1420が「2」であった場合を想定すると、暗号化判定部2110は、暗号化情報記憶部2800から暗号化度情報2810を読み出し、セキュリティ強度2811が「2」に対応するデータ暗号化強度2812が「強」であることを読み出す。

このデータ暗号化強度2812が「強」の暗号化方法を実行装置3000がサポートしている場合、読み出したセキュリティ強度情報1420に対応する暗号化機能を実行装置3000が備えていると判断する。尚、本実行装置において、どのレベルの暗号化がサポートされているかは、予め決められており、アプリ登録部2100は記憶しているものとする。

[0071] 実行装置3000が、読み出したセキュリティ強度情報1420に対応する暗号化機能を備えていると暗号化判定部2110が判断した場合には(ステップS930:YES)、アプリ登録部2100は、アプリケーションをロードするためのクラスローダオブジェクトを生成する(ステップS940)。

次に、読み出したセキュリティ強度情報1420に応じて、暗号アルゴリズムと暗号化に使用する鍵の長さを決定し、暗号化鍵を生成する(ステップS950)。暗号鍵は、同じアプリケーションであっても、登録するたびにランダムに生成することとする。より判読されにくくなるからである。また、セキュリティ強度情報1420で指定されたよりも強い暗号アルゴリズムや鍵を用いてもよいものとする。

[0072] アプリ登録部2100は、クラスローダID2821、クラスローダオブジェクトのアドレス2

822、暗号アルゴリズム2823、暗号鍵2824とを対応づけて、暗号アルゴリズム情報2820へ登録する(ステップS960)。

セキュリティ強度情報1420に対応する暗号化機能を実行装置3000が備えていない場合には、アプリケーションの起動を中断する(ステップS970)。アプリケーションを起動しないことで、アプリケーションの秘匿性が守られることになる。

[0073] <2. オブジェクト生成処理>

図10を用いて説明する。図10は、オブジェクトの生成処理を表すフローチャートである。

クラスローダオブジェクトによる起動クラスのロードが完了する(図8、ステップS860参照)と、インタプリタ2300は、起動クラスのメソッドを実行することでアプリケーションの起動を行なう。

[0074] メソッドを実行するとは、まず、ヒープ管理部2400に、オブジェクトの作成を依頼する。

依頼を受けたヒープ管理部2400は、新しいオブジェクトのためのメモリ領域を確保する(ステップS1010)。

次に、インタプリタ2300は、カレントクラスがセキュアクラスであるかを検査する。これはカレントクラスのセキュアフラグ1217を調べることで判定できる。セキュアフラグ1217が「ON」であれば、セキュアクラス、すなわち、暗号化が必要なクラスであると判定する。

[0075] カレントクラスがセキュアクラスと判定された場合には(ステップS1020:YES)、オブジェクトヘッダ2511内の暗号化フラグにオブジェクトが暗号化されていることを表す「1」を設定する(ステップS1040)。

フラグの設定後、オブジェクトデータ2512を暗号化する(ステップS1040)。ここでの暗号化の方法は、暗号アルゴリズム情報2820(図6参照)を読み出すことで、決定する。

[0076] カレントクラスがセキュアクラスではないと判定された場合には(ステップS1020:NO)、オブジェクトヘッダ2511内の暗号化フラグにオブジェクトが暗号化されていないことを表す「0」を設定する(ステップS1050)。

ここで、カレントクラスとは、実行中のメソッドを定義しているクラスをいう。

インタプリタ2300は、Javaメソッドを実行する際にJavaフレームと呼ばれるデータ構造をRAM3600上に作成する。Javaフレームは Javaメソッドの呼び出し毎に一つ生成され、メソッドの実行が終了した時に破壊される。インタプリタ2300は、複数のスレッドにより実行されるが、制御権が与えられているスレッドの任意の場所で、実行中のメソッドに対してアクティブなJavaフレームは一つしか存在しない。このフレームのことをカレントフレームと呼び、そこで実行中のメソッドをカレントメソッドと呼ぶ。そのカレントメソッドを定義しているクラスがカレントクラスである。

[0077] すなわち、カレントクラスは、ある瞬間には1つに定まる。

＜補足＞

以上、本発明に係る実行装置について実施形態に基づいて説明したが、この実行装置を部分的に変形することもでき、本発明は上述の実施形態に限られないことは勿論である。即ち、

(1)実施形態では、実行装置3000で実行するアプリケーションは、アプリ取得プログラム3210が、本装置外のアプリケーションファイル1000からダウンロードしたものであることとしているが、インターネット上にあるサーバから、ダウンロードすることとしてもよい。

[0078] この場合、アプリ取得プログラム3210は、TLS(Transport Layer Security)、HTTP(Hypertext Transfer Protocol)等のプロトコルに従いJavaアプリケーションをダウンロードする機能を有するプログラムとなる。

ここで、TLSは暗号化により通信時のデータの盗聴、改竄を防ぐデータ転送方式である(RFC2246参照)。また、HTTPは、インターネット上のデータ通信で一般的に用いられているデータ転送方式である(RFC2616参照)。

[0079] 尚、RFC(Request For Comments)とは、インターネット上の技術を規格化するIETF(Internet Engineering Task Force)の公式文書であり、プロトコルなど様々な技術の使用がまとめられているものである。

また、実行装置3000で実行するアプリケーションは、デジタル放送のデータ放送として、MPEG(Moving Picture Coding Experts Group)2トランスポートストリーム内に

埋め込まれたJavaアプリケーションであってもよい。

[0080] この場合、アプリ取得プログラム3210は、トランスポートストリーム内に埋め込まれたJavaアプリケーションを実行装置3000内に読み出すプログラムとなる。

MPEG2トランスポートストリームにJavaプログラムを埋め込む方法としては、例えば、DSMCC方式がある。DSMCC方式とは、MPEG2トランスポートストリームのパケットの中に、コンピュータで使用されているディレクトリやファイルで構成されるファイルシステムをエンコードする方法である(MPEG規格書ISO／IEC138181－1、MPEG規格書ISO／IEC138181－6参照)。

[0081] またさらに、実行装置3000で実行するアプリケーションは、SDカード(Secure Digital memory card)、CD－ROM(Compact Disk Read Only Memory)、DVD(Digital Versatile Disk)、Blu－RayDisc等に記録されたJavaアプリケーションであってもよい。

この場合、アプリ取得プログラム3210は、これらの記録媒体からアプリケーションを読み出すプログラムとなる。

[0082] また、実行装置3000で実行するアプリケーションは、実行装置3000内にあるROM3500に記録されたJavaアプリケーションであってもよい。

この場合、アプリ取得プログラム3210は、ROM3500からRAM3600に、Javaアプリケーションを読み出すプログラムとなる。

(2)本実施形態ではアプリ取得プログラム3210などはJava言語で記述されたJavaプログラムとしているが、同等の機能を有する、ネイティブ言語で記述されたプログラムや、ハードウェアで実現されていてもよい。

[0083] また、JAVAバーチャルマシンで実行するアプリケーションは、Java言語で記述されたものに限らず、C＋＋などの他のオブジェクト思考言語で記述されたものであってもよい。

(3)本実施形態では、セキュリティ強度2811は「0」～「2」の3段階としているが、これに限るものではない。

[0084] 例えば、4段階以上にセキュリティ強度を設定してもよいし、データの暗号化が必要、不要の2段階であってもよい。

(4)本実施形態では、アプリケーション1001のメタデータ1400は、起動クラス名1410とセキュリティ強度情報1420とを含むこととしているが、これら以外の情報を含んでいてもよい。

[0085] また、セキュリティ強度を指定する代わりに、メタデータ1400内で暗号アルゴリズムや鍵の長さを指定することとしてもよい。

なお、アプリケーション1001がメタデータ1400を含まないこととする場合には、暗号化判定部2110は、そのアプリケーションのセキュリティ強度情報を「0」とみなしたり、実行装置3000であつかえる最も高いレベルのセキュリティ強度とみなしたりすることとしてもよい

(5)本実施形態では、アプリケーション1001は、一つのファイルとして構成されているが、アプリケーションクラス1200、データファイル1300、メタデータ1400が別々のファイルであってもよい。また、メタデータ1400をアプリケーションクラス1200に埋め込むこととしてもよい。

(6)実施形態では、アプリケーション1001のセキュリティ強度情報1420によって、クラスのセキュアフラグ1217が一律に決まることとしているが、このセキュアフラグ1217は、クラスごとに設定できることとしてもよい。

[0086] 例えば、暗号化が必要なクラスのセキュアフラグ1217は「ON」とし、暗号化が不要なクラスは「OFF」とするなどである。また、クラスごとに、セキュリティ強度情報1420を設定できることとしてもよい。

(7)本実施形態においては、オブジェクトデータ2512全体を暗号化しているが、フィールドをそれぞれ個別に暗号化してもよい。

[0087] また、フィールドを暗号化する際に、基本型のフィールドのみを暗号化し、参照型のフィールドの暗号化を行わなくしてもよい。この場合、参照型フィールドを識別できない場合には、フィールドごとに暗号化されているか否かのフラグを持つ必要がある。

これにより、多くの参照型のフィールドへのアクセスを必要とするヒープ管理部2400が行うガベージコレクションを高速に実行できる。

(8)実施形態で示した実行装置の各機能を実現させる為の各制御処理(図2等参照)をCPUに実行させる為のプログラムを、記録媒体に記録し又は各種通信路等を介

して、流通させ頒布することもできる。このような記録媒体には、ICカード、光ディスク、フレキシブルディスク、ROM、フラッシュメモリ等がある。流通、頒布されたプログラムは、機器におけるCPUで読み取り可能なメモリ等に格納されることにより利用に供され、そのCPUがそのプログラムを実行することにより実施形態で示した実行装置の各機能が実現される。

[0088] ＜従来技術の説明＞

図11に示すように、この従来のデータ保護機能を備えた計算機は、中央処理装置203の内部に、データの復号処理を行う暗号化データ解読手段204、暗号化処理を行うデータ暗号化手段205、データバッファ206、プログラム実行手段207とを備え、中央処理装置203の外部にある暗号化データ202を中央処理装置203内のデータバッファ206へ読み込む際に復号し、プログラム実行手段207で処理した後に、データ暗号化手段205により暗号化して中央演算装置の外部へと出力している。

産業上の利用可能性

[0089] Javaアプリケーションが実行時に生成するデータの盗聴、改ざんから保護することができ、今後本格展開が予想されるJavaアプリケーションのダウンロード配信ビジネスにおいて、コンテンツ作成者の権利を保護すること場合などに、特に有用である。

ダウンロード配信ビジネスとして、例えば、携帯電話機では、NTT DoCoMoがi－アプリと呼ばれるサービスを提供している。このサービスは、携帯電話端末がインターネット上にあるアプリケーション配信サーバからJavaプログラムをダウンロードして、端末上で実行する。また、欧州では、DVB－MHP（Digital Video Broadcasting －Multimedia Home Platform）と呼ばれる仕様が策定され、既に仕様に準拠した運用が開始されている。DVB－MHP規格に基づくデジタル放送では、放送波に多重化されたJavaプログラムをデジタルTVが受信し、それを実行する。

請求の範囲

[1] オブジェクト指向言語で作成されたアプリケーションプログラムを実行する実行装置であって、

前記アプリケーションプログラムは、1以上のメソッドを有する1以上のクラスと秘匿の必要性の有無を表す秘匿性情報とを含んでおり、

前記秘匿性情報に基づいて、暗号化が必要であるか否かを判断する暗号化判断手段と、

前記メソッドを実行する際に、前記メソッドが操作するデータを含むオブジェクトをメモリに記録するオブジェクト記録手段とを備え、

前記暗号化判断手段で暗号化が必要であると判断された場合においては、前記オブジェクト記録手段は、暗号化されたデータを含むオブジェクトを記録する

ことを特徴とする実行装置。

[2] 前記秘匿性情報は、更に、秘匿の度合いを示す情報を含み、

前記実行装置は、更に、前記秘匿性情報に基づいて暗号化方式を決定する暗号方式決定手段を備え、

前記暗号化判断手段で暗号化が必要であると判断された場合においては、前記オブジェクト記録手段は、前記暗号方式決定手段で決定された暗号化方式で暗号化されたデータを含むオブジェクトを記録する

ことを特徴とする請求項1記載の実行装置。

[3] 前記オブジェクト内のデータを書き換えるときは、前記データが暗号化されている場合は、暗号化されたデータを記録する

ことを特徴とする請求項1記載の実行装置。

[4] 前記メモリに記録されたオブジェクトは、オブジェクト内のデータが暗号化されているか否かを示す情報を有し、

前記情報がオブジェクト内のデータが暗号化されていることを示している場合は、暗号化されたデータを記録する

ことを特徴とする請求項3記載の実行装置。

[5] 前記実行装置は、更に、データが他のデータの所在位置特定に必要なデータであ

るか否かを判別する判別手段を備え、

前記判別手段が、他のデータの所在位置特定に必要なデータであると判別したデータの場合は、暗号化を抑止する

ことを特徴とする請求項1記載の実行装置。

[6] プログラムを実行する実行装置であって、

前記プログラムは、データ部と秘匿の必要性の有無を表す秘匿性情報とを含んでおり、

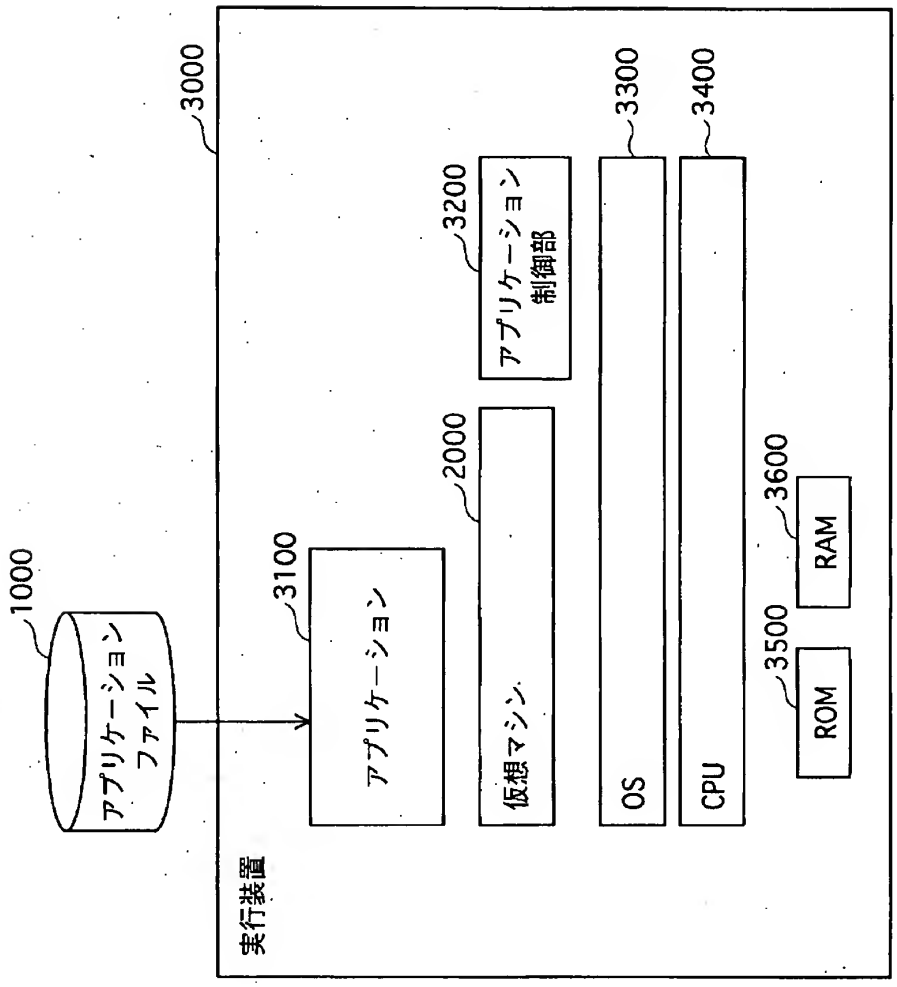
前記秘匿性情報に基づいて、暗号化が必要であるか否かを判断する暗号化判断手段と、

前記実行可能プログラムを実行する際に、前記データ部をメインメモリにロードするロード手段とを備え、

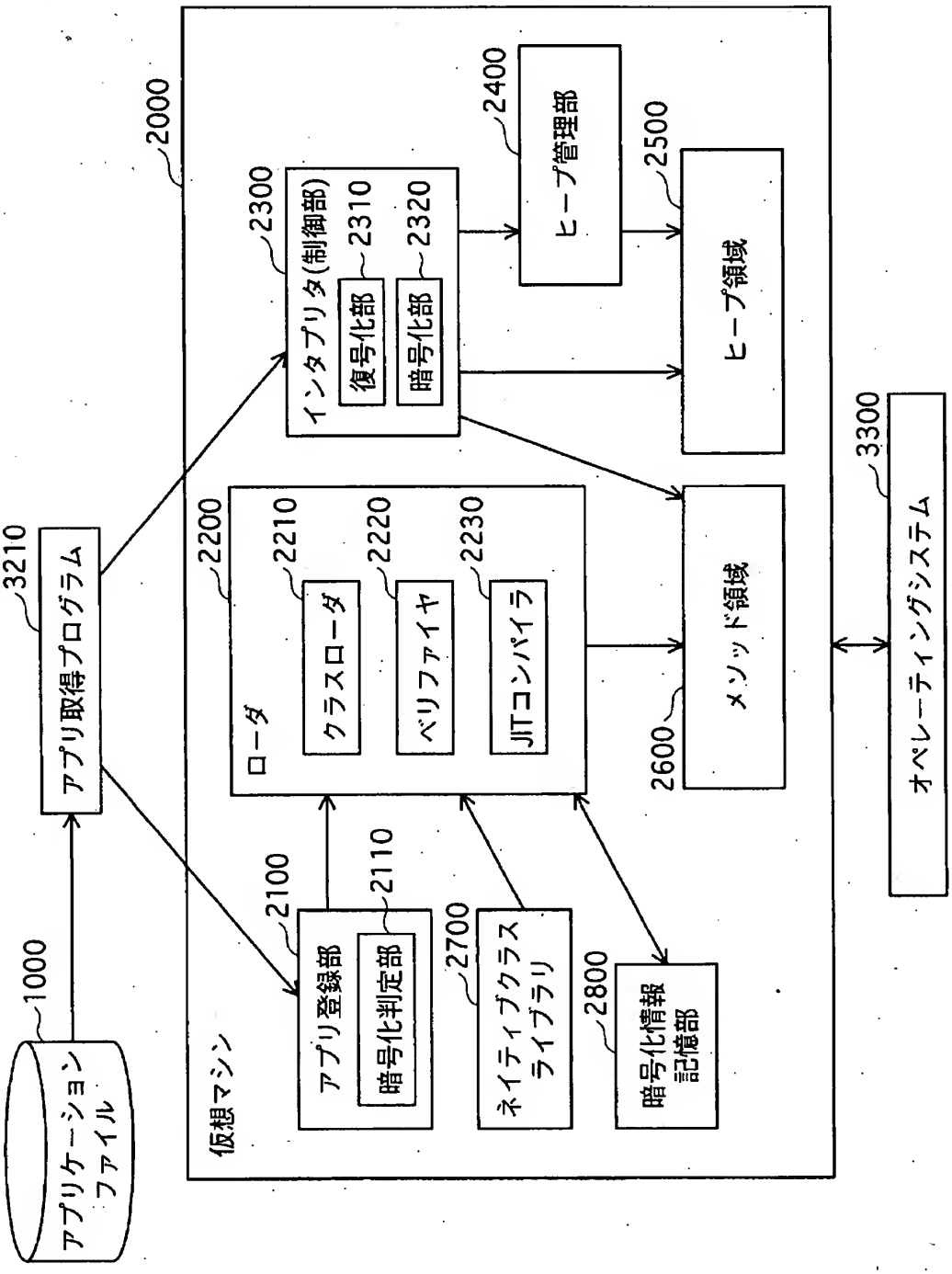
前記ロード手段は、前記暗号化判断手段で暗号化が必要であると判断された場合は、暗号化してデータ部をロードする

ことを特徴とする実行装置。

[図1]


3000
3200
アプリケーション制御部
3300
3400
2000
仮想マシン
OS
CPU
3600
RAM
3500
ROM
1000
アプリケーションファイル
3100
アプリケーション
実行装置

[図2]

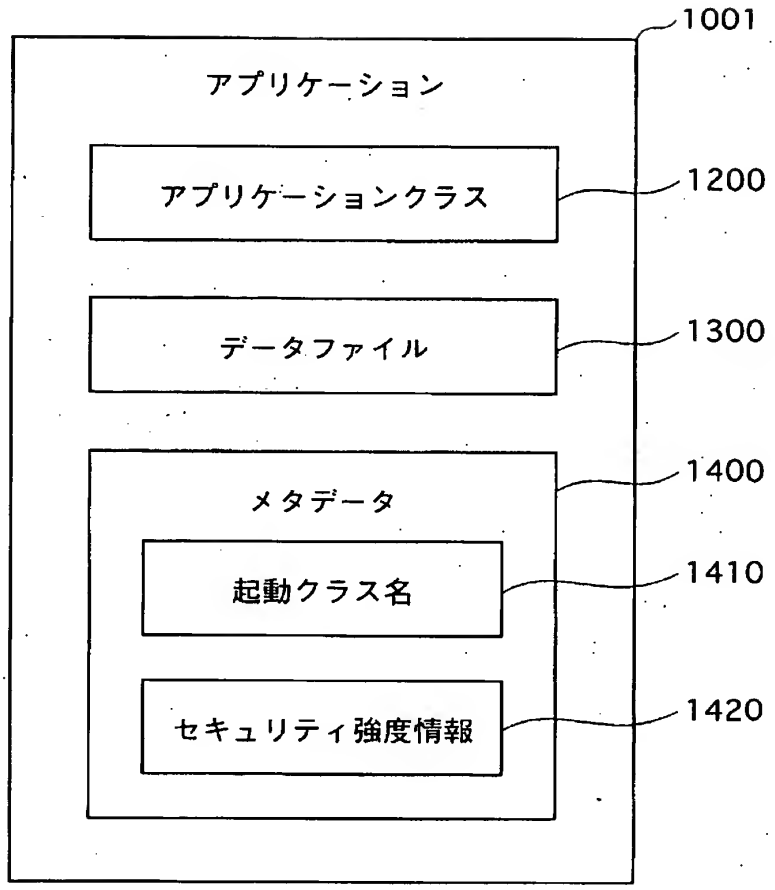

2000
仮想マシン
1000
アプリケーション
ファイル
3210
アプリ取得プログラム
2100
アプリ登録部
2110
暗号化判定部
2200
ローダ
2210
クラスローダ
2220
ベリファイヤ
2230
JITコンパイラ
2300
インタプリタ(制御部)
2310
復号化部
2320
暗号化部
2400
ヒープ管理部
2500
ヒープ領域
2600
メソッド領域
2700
ネイティブクラス
ライブラリ
2800
暗号化情報
記憶部
3300
オペレーティングシステム

[図3]

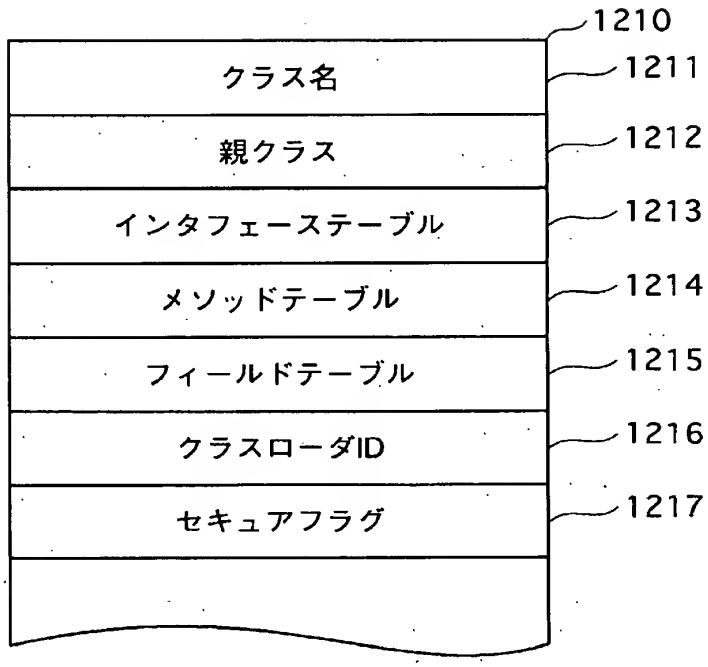
2811 2812 2810

セキュリティ強度	データ暗号化強度
0	不要
1	弱
2	強

[図4]


1001
アプリケーション
アプリケーションクラス
1200
データファイル
1300
メタデータ
1400
起動クラス名
1410
セキュリティ強度情報
1420

[図5]

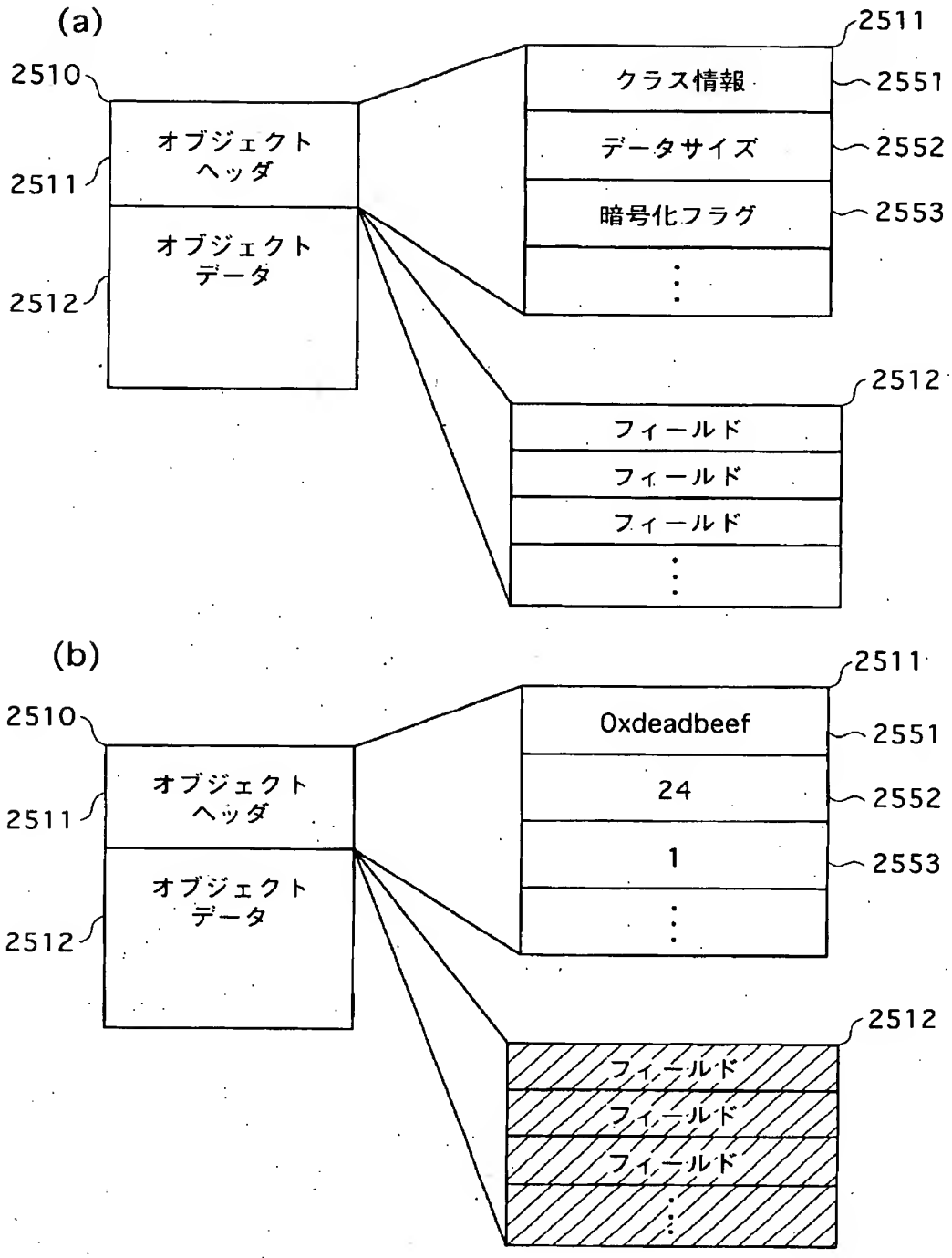

1210
クラス名
1211
親クラス
1212
インタフェーステーブル
1213
メソッドテーブル
1214
フィールドテーブル
1215
クラスローダID
1216
セキュアフラグ
1217

[図6]

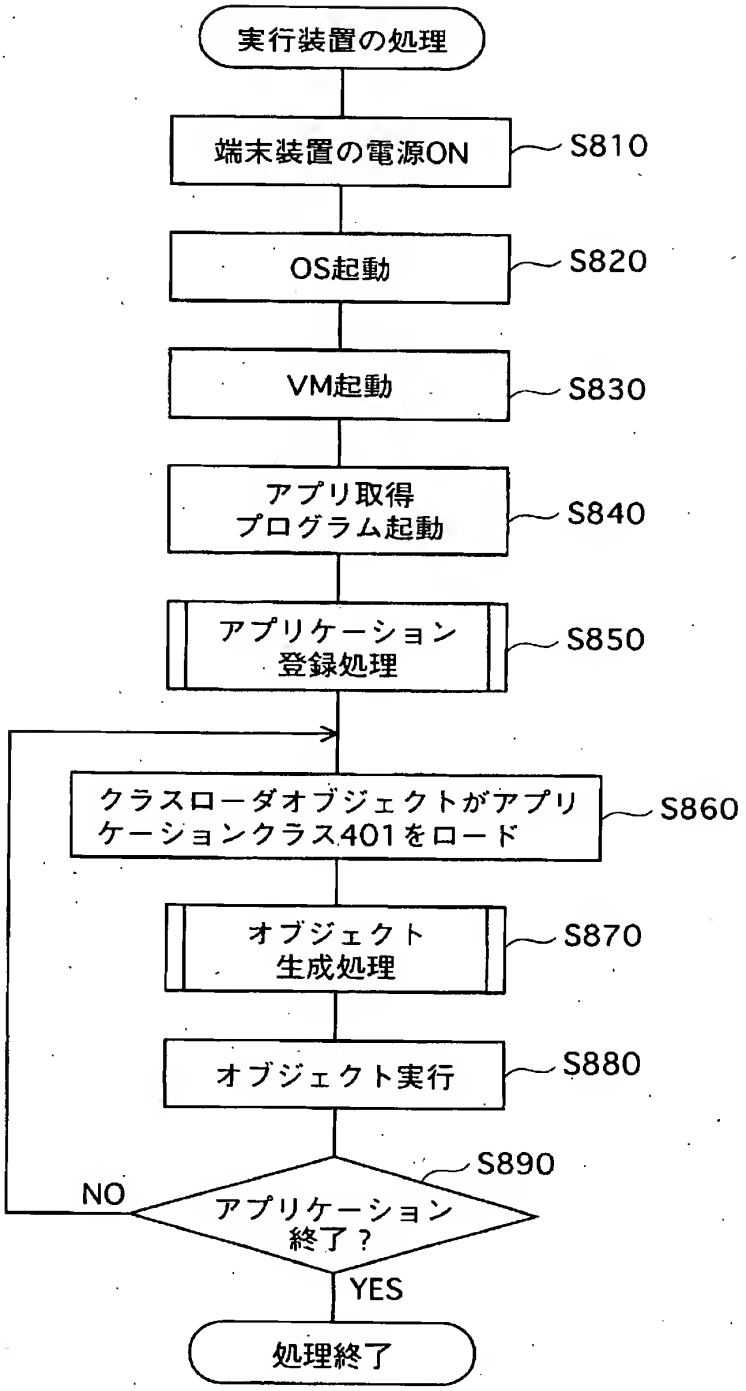
2820

2821 クラスローダID	2822 クラスローダアドレス	2823 暗号アルゴリズム	2824 暗号鍵
0	0x30000000	-	-
1	0x30000010	DES	-
2	0x30000020	AES	YYYY

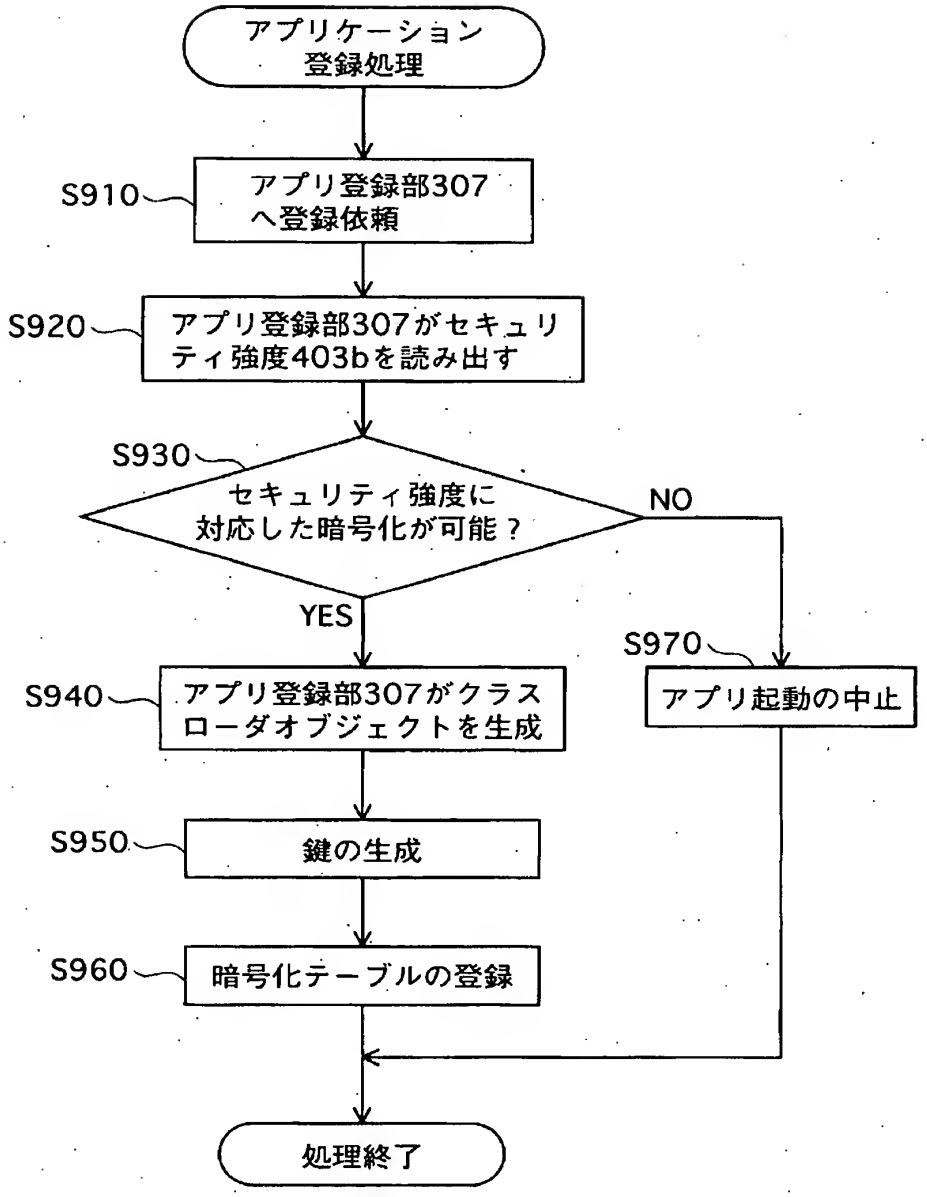
[図7]


(a)
2510
2511
2512
オブジェクト
ヘッダ
オブジェクト
データ
2511
クラス情報
2551
データサイズ
2552
暗号化フラグ
2553
2512
フィールド
フィールド
フィールド
(b)
2510
2511
2512
オブジェクト
ヘッダ
オブジェクト
データ
2511
0xdeadbeef
2551
24
2552
1
2553
2512
フィールド
フィールド
フィールド

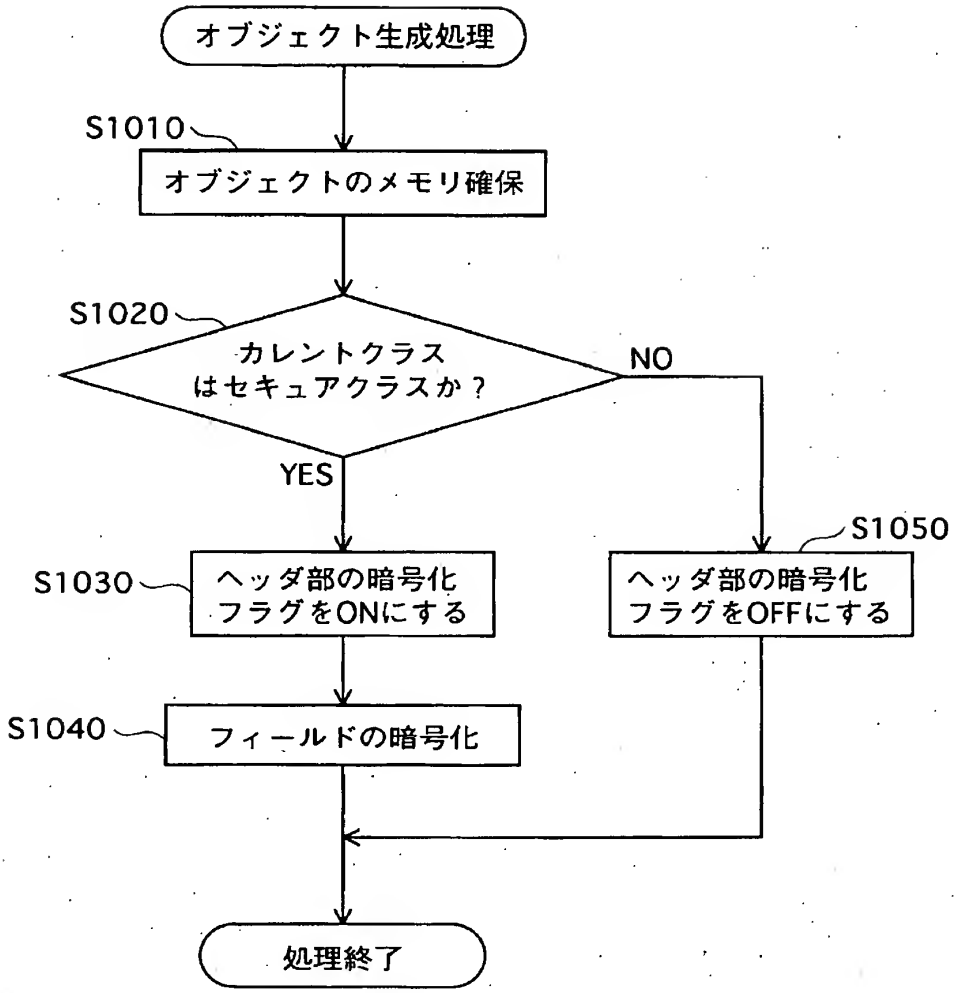
[図8]


実行装置の処理
端末装置の電源ON
S810
OS起動
S820
VM起動
S830
アプリ取得
プログラム起動
S840
アプリケーション
登録処理
S850
クラスローダオブジェクトがアプリ
ケーションクラス401をロード
S860
オブジェクト
生成処理
S870
オブジェクト実行
S880
S890
NO
アプリケーション
終了？
YES
処理終了

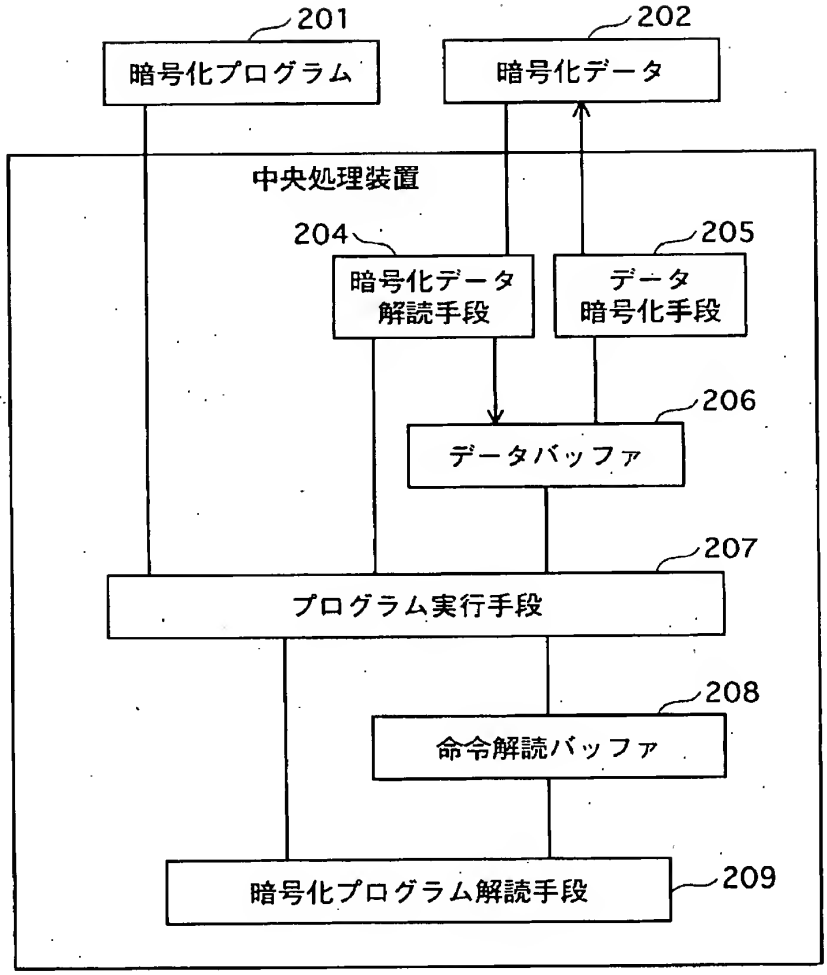
[図9]


アプリケーション
登録処理
S910
アプリ登録部307
へ登録依頼
S920
アプリ登録部307がセキュリ
ティ強度403bを読み出す
S930
セキュリティ強度に
対応した暗号化が可能？
NO
YES
S970
アプリ起動の中止
S940
アプリ登録部307がクラス
ローダオブジェクトを生成
S950
鍵の生成
S960
暗号化テーブルの登録
処理終了

[図10]


オブジェクト生成処理
S1010
オブジェクトのメモリ確保
S1020
カレントクラス
はセキュアクラスか？
NO
YES
S1050
S1030
ヘッダ部の暗号化
フラグをONにする
ヘッダ部の暗号化
フラグをOFFにする
S1040
フィールドの暗号化
処理終了

[図11]


201
暗号化プログラム
202
暗号化データ
中央処理装置
204
暗号化データ
解読手段
205
データ
暗号化手段
206
データバッファ
207
プログラム実行手段
208
命令解読バッファ
暗号化プログラム解読手段
209

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/006290

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] G06F1/00, 9/44, 12/14, H04L9/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[7] G06F1/00, 9/44, 12/14, H04L9/14

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2005
Kokai Jitsuyo Shinan Koho 1971-2005 Toroku Jitsuyo Shinan Koho 1994-2005

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2001-350664 A (Nippon Telegraph And Telephone Corp.), 21 December, 2001 (21.12.01), Full text; all drawings (Family: none)	1-5
Y	JP 2003-345664 A (Nissan Motor Co., Ltd.), 05 December, 2003 (05.12.03), Claims (Family: none)	1-5
Y	JP 02-155034 A (Toshiba Corp.), 14 June, 1990 (14.06.90), Full text; all drawings (Family: none)	6
A		1-5

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 June, 2005 (24.06.05)	12 July, 2005 (12.07.05)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/006290

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	JP 2002-366437 A (Sharp Corp.), 20 December, 2002 (20.12.02), Claim 6 & US 2002/184495 A1	6 1-5
A	WO 2002/003208 A2 (INTEL CORP.), 10 January, 2002 (10.01.02), Full text; all drawings & JP 2004-523015 A & EP 1314091 A	1-6
A	JP 2003-290989 A (Toshiba Corp.), 03 October, 2003 (03.10.03), Full text; all drawings & US 2003/182571 A1 & EP 1347384 A2	1-6
A	Java TM Cryptography Extension (JCE) Reference Guide for the Java TM 2 SDK, Standard Edition, v.1.4.[online].Sun Microsystems Inc, 10 January, 2002 (10.01.02), [retrieved on 24 June, 2005 (24.06.05)]., Reterieved from the Internet:<URL:http://java.sun.com/j2se/ 1.4.2/docs/guide/security/jce//JCERefGuide. html>	1-6

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００５／００６２９０

A.　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.[7]　G06F1/00, 9/44, 12/14, H04L9/14

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.[7]　G06F1/00, 9/44, 12/14, H04L9/14

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００５年
日本国実用新案登録公報	１９９６－２００５年
日本国登録実用新案公報	１９９４－２００５年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C.　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Y	JP 2001-350664 A(日本電信電話株式会社) 2001.12.21, 全文, 全図 (ファミリーなし)	1-5
Y	JP 2003-345664 A(日産自動車株式会社) 2003.12.05, 特許請求の範囲 (ファミリーなし)	1-5
Y	JP 02-155034 A(株式会社東芝) 1990.06.14, 全文, 全図 (ファミリーなし)	6
A		1-5

☑ Ｃ欄の続きにも文献が列挙されている。　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日 ２４．０６．２００５	国際調査報告の発送日 12.7.2005
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員）　５Ｓ　９５５５ 宮司　卓佳 電話番号　０３－３５８１－１１０１　内線　３５４６

様式ＰＣＴ／ＩＳＡ／２１０（第２ページ）（２００４年１月）

C（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Y	JP 2002-366437 A(シャープ株式会社) 2002.12.20,請求項６	６
A	& US 2002/184495 A1	1—5
A	WO 2002/003208 A2 (INTEL CORPORATION) 2002.01.10,全文,全図 & JP 2004-523015 A & EP 1314091 A	1—6
A	JP 2003-290989 A (株式会社東芝) 2003.10.03,全文,全図 & US 2003/182571 A1 & EP 1347384 A2	1—6
A	JavaTM Cryptography Extension (JCE) Reference Guide for the JavaTM 2 SDK, Standard Edition, v.1.4. [online]. Sun Microsystems Inc, 2002.01.10. [retrieved on 2005-06-24]. Retreived from the Internet:<URL: http://java.sun.com/j2se/1.4.2/docs/guide/security/jce//JCERefGuide.html>	1—6